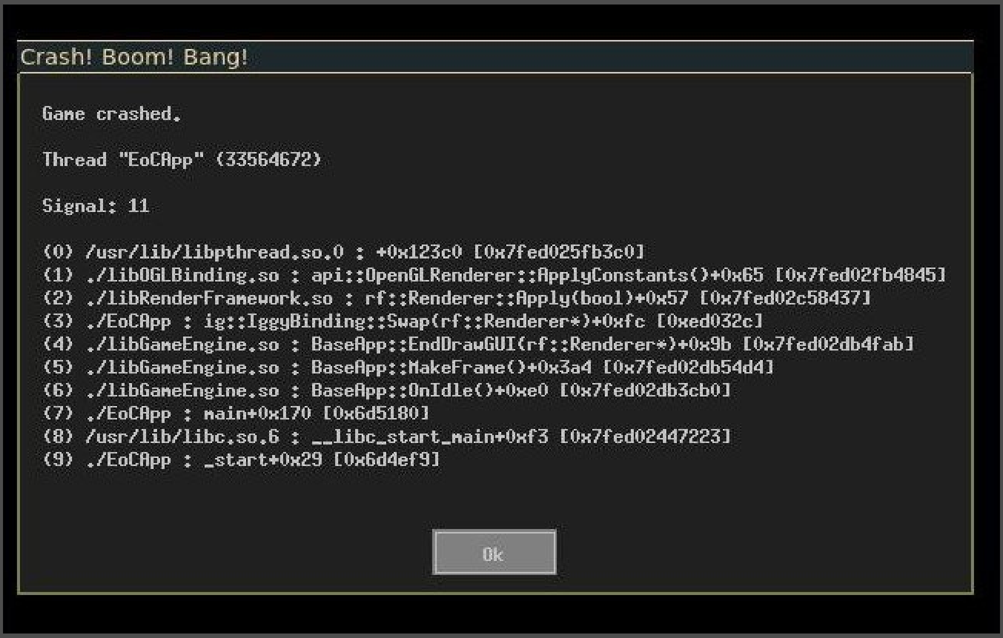
Articles

Divinity Manjaro Game Crash Fix

Introduction



```
Crash! Boom! Bang!

Game crashed.

Thread "EoCApp" (33564672)

Signal: 11

(0) /usr/lib/libpthread.so.0 : +0x123c0 [0x7fed025fb3c0]
(1) ./libOGLBinding.so : api::OpenGLRenderer::ApplyConstants()+0x65 [0x7fed02fb4845]
(2) ./libRenderFramework.so : rf::Renderer::Apply(bool)+0x57 [0x7fed02c58437]
(3) ./EoCApp : ig::IggyBinding::Swap(rf::Renderer*)+0xfc [0xed032c]
(4) ./libGameEngine.so : BaseApp::EndDrawGUI(rf::Renderer*)+0x9b [0x7fed02db4fab]
(5) ./libGameEngine.so : BaseApp::MakeFrame()+0x3a4 [0x7fed02db54d4]
(6) ./libGameEngine.so : BaseApp::OnIdle()+0xe0 [0x7fed02db3cb0]
(7) ./EoCApp : main+0x170 [0x6d5180]
(8) /usr/lib/libc.so.6 : __libc_start_main+0xf3 [0x7fed02447223]
(9) ./EoCApp : _start+0x29 [0x6d4ef9]
```

This week I got the computer game Divinity: Original Sin Enhanced Edition as a gift from a friend on Steam. Unfortunately the game crashed on my Manjaro OS when I started it and I had to find a solution from Reddit and GOG forum on the internet. Here I describe the summary for the fix.

Realisation

We open a terminal and go with `~/.local/share/Steam/steamapps/common/Divinity Original Sin Enhanced Edition` into the installation folder of the computer game. In this folder we create the file `divos_hack.c` and copy the following source code into it.

```
/*
 * LD_PRELOAD shim which applies two patches necesary to get the game
 * Divinity: Original Sin Enhanded Edition for Linux to work with Mesa (12+)
 *
 * Build with: gcc -s -O2 -shared -fPIC -o divos-hack.{so,c} -ldl
 */

/* for RTLD_NEXT */
#ifndef _GNU_SOURCE
#define _GNU_SOURCE
#endif
#include
#include
#include

#define _GLX_PUBLIC

/*
 *   https://github.com/karolherbst/mesa/commit/aad2543bf6cfbd7df795d836e5ff4ec8686e4fdf
 *     - allow env override of vendor string.  I actually just hard-coded
 *       ATI Technologies, Inc., since that appears to be what's needed
 */
const GLubyte *GLAPIENTRY glGetString( GLenum name )
{
    static void *next = NULL;
    static const char *vendor = "ATI Technologies, Inc.";
    if(name == GL_VENDOR)
 return (const GLubyte *)vendor;
    if(!next)
 next = dlsym(RTLD_NEXT, "glGetString");
    return ((const GLubyte *GLAPIENTRY (*)(GLenum))next)(name);
}

/*
 *   https://gist.github.com/karolherbst/b279233f8b13c9db1f3e1e57c6ecfbd2
 */
_GLX_PUBLIC void (*glXGetProcAddressARB(const GLubyte * procName)) (void)
{
   static void *next = NULL;
   if (strcmp((const char *) procName, "glNamedStringARB") == 0 ||
       strcmp((const char *) procName, "glDeleteNamedStringARB") == 0 ||
       strcmp((const char *) procName, "glCompileShaderIncludeARB") == 0 ||
       strcmp((const char *) procName, "glIsNamedStringARB") == 0 ||
       strcmp((const char *) procName, "glGetNamedStringARB") == 0 ||
       strcmp((const char *) procName, "glGetNamedStringivARB") == 0)
      return NULL;
   if(!next)
 next = dlsym(RTLD_NEXT, "glXGetProcAddressARB");
    return ((_GLX_PUBLIC void (*(*)(const GLubyte *))(void))next)(procName);
}
```

We save the file and compile it with `gcc -s -O2 -shared -fPIC -o divos-hack.{so,c} -ldl`. The result is the `divos-hack.so` file. Now we open the `runner.sh` file and enter the following source code.

```
#!/bin/sh

HOME=${HOME}/.local/share

LD_PRELOAD="divos-hack.so"
LD_LIBRARY_PATH="." ./EoCApp
```

After that we start the game normally over the Steam launcher.